[0044] According to an embodiment, Sb and Sr may be calculated as:

$$S_b = \frac{1}{\frac{1-M[1,3]}{2}} \text{ and } S_r = \frac{1}{\frac{1-M[1,1]}{2}}.$$

[0045] FIG. **3** is a block diagram of an example color format conversion for encoding. Once the color space conversion is completed, quantization and subsampling operations are conventionally performed on the image data in the Y'CbCr color space prior to delivering the data onto an encoder or pre-processing system. In FIG. **3** color space conversion can be performed as described above in converter **302**, followed by quantizer **204** and downsampler **306**. According to an embodiment, the quantized value of Y' (Yq) may be calculated as Qy(Y') where Qy represents the quantization function. If a downsampling process is used, e.g. from 4:4:4 to 4:2:2 or 4:2:0, or from 4:2:2 to 4:2:0, the down sampled version of Cb and Cr need to be generated using a particular downsampling process. This process can be represented as Down(QCb(Cb)) and Down(QCr(Cr)) respectively where Down( ) is the down sampling function. These steps are commonly inverted prior to display or some other processing that may be desirable. After these processes are completed, and if these data are then passed into an encoder then encoding processes are performed directly on such data. For example, the output of downsampler **306** may be input to pre-processor **104** or directly into coding engine **108** where, for example, prediction, DCT or waveform transform, quantization, and other coding processes may be performed.

[0046] Similar functions can be performed inversely as part of decoding operations to generate reconstructed Br', Rr', and Gr'. For example, Yr' may be calculated as IQY (YqN) where IQ( ) is the inverse quantization function. IqN is essentially Iq+N, where N is a noise that is added during encoding or transmission of the data (e.g. quantization noise). The Cbr and Crr may be calculated as IQCb(UP (CbqN)) and IQCr(UP(CrqN)) respectively where UP( ) is an up-sampling function and CbqN and CrqN are the reconstructed chroma data with, similarly some coding noise also introduced into them due to the coding process.

[0047] Then, for the conventional method, Br' may be calculated as Cbr/Sb+Yr' and Rr' may be calculated as Crr/Sr+Yr'. Then, solving the above equation for the calculation of Y' for the value of G', Gr' may be calculated as (Ye−M[1,1]*Re−M[1,3]*Br')/M[1,2].

[0048] In the methods presented above, the presented conversion and encoding operations are inverted at the decoder. For example, in the constant luminance case, the inversion process for generating B, R, and G components is now performed as follows:

$$Cbr\_\text{temp}=\text{sign}(Cbr)*ITFcb(\text{abs}(Cbr))$$

$$Crr\_\text{temp}=\text{sign}(Crr)*ITFcr(\text{abs}(Crr))$$

$$Br=Cbr\_\text{temp}/Sb+Y$$

$$Rr=Crr\_\text{temp}/Sr+Y$$

$$Gr=(Yr-M[1,1]*Rr-M[1,3]*Br)/M[1,2]$$

Similarly, for the non-constant luminance method:

$$Cbr\_\text{temp}=\text{sign}(Cbr)*ITFcb(\text{abs}(Cbr))$$

$$Crr\_\text{temp}=\text{sign}(Crr)*ITFcr(\text{abs}(Crr))$$

$$Br=Cbr\_\text{temp}/Sb+ITF(Y')$$

$$Rr=Crr\_\text{temp}/Sr+ITF(Y')$$

$$Gr=(Yr-M[1,1]*Rr-M[1,3]*Br)/M[1,2]$$

[0049] Similar considerations for deriving Sb and Sr need to be applied as previously discussed, i.e., we need to ensure that the Cb and Cr quantities remain within the [−0.5, 0.5] range.

[0050] Adaptive Chroma Downsampling

[0051] As mentioned above, an important step for preparing content for compression is the conversion of the original video data that may be in a 4:4:4 or 4:2:2 arrangement, into a reduced chroma resolution space using a chroma downsampling method. Given the characteristics of the chroma signal, such a step could improve compression while also reducing complexity since fewer samples need to be processed during encoding. An adaptive chroma downsampling method, e.g. from 4:4:4 to 4:2:2 or 4:2:0 is described that involves the consideration of N possible downsampling filters, and adaptively selects the “best” filter for each sample position, resulting in improved performance.

[0052] A 4:2:2 format is a luminance-chrominance format where the chrominance channels are downsampled in one dimension, containing a single luminance value for every pixel, but each of the two chrominance channels contain only one value for every other pixel along one dimension of the image. This results in each of the two chrominance channels in a 4:2:2 format containing only one value for every two luma samples along one (typically the horizontal) dimension of the image. A 4:2:0 format is a luminance-chrominance format where the chrominance channels are downsampled in both dimensions, each of the chrominance channels containing only one value for every two luma samples along both dimensions of the image. This results in each chrominance channel in a 4:2:2 format containing a quarter as many values as the luminance channel. A 4:4:4 format is not downsampled and includes three color component values for every pixel.

[0053] FIG. **3** is a block diagram of an example color format conversion for encoding. The color format converter of FIG. **3** may be incorporated as part of the pre-processor **104** of FIG. **1**, and may output data in a format appropriate to the coding engine **108**. In this example, source video **102** is converted in color space, for example from RGB to any of the 4:4:4 luminance-chrominance formats described herein. Downsample filter selector **204** chooses a chroma downsample filter from a set of downsample filter options as described below regarding FIG. **4**. The chosen downsample filter **305** is used to generate pre-processed data **202** from the chosen downsample filter **305**.

[0054] FIG. **4** is a flowchart for an example process for choosing a downsample filter. According to an embodiment, the decision of which filter to use for each sample position is based on examining the performance of each filter given a known upsampling filter that would likely be used after decoding (e.g. prior to display, delivery through an interface system etc.). Each filter is evaluated and the one resulting in the least distortion, e.g. sum of square errors (SSE), sum of